FIG. 1

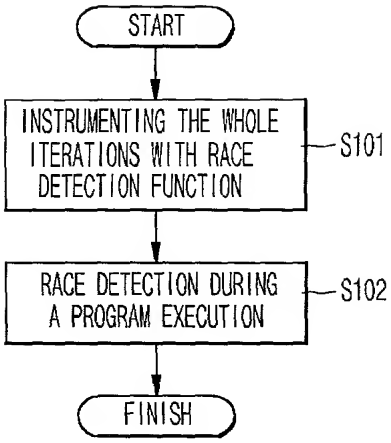

START
INSTRUMENTING THE WHOLE ITERATIONS WITH RACE DETECTION FUNCTION
S101
RACE DETECTION DURING A PROGRAM EXECUTION
S102
FINISH

FIG. 2

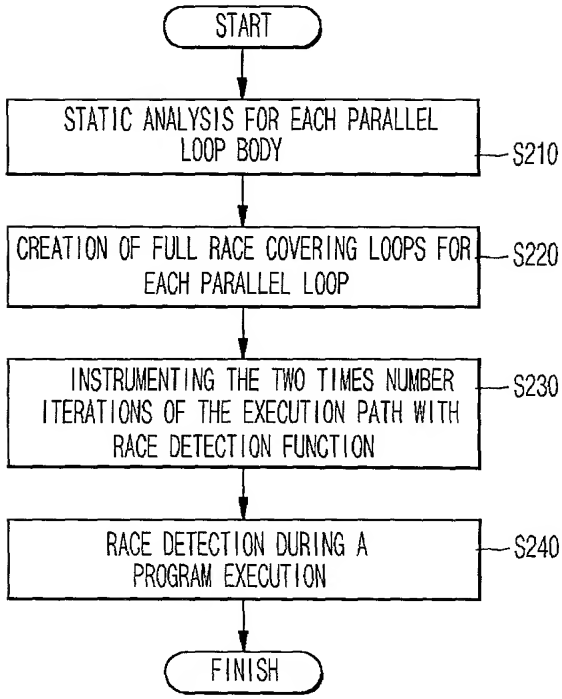

START
STATIC ANALYSIS FOR EACH PARALLEL LOOP BODY
S210
CREATION OF FULL RACE COVERING LOOPS FOR EACH PARALLEL LOOP
S220
INSTRUMENTING THE TWO TIMES NUMBER ITERATIONS OF THE EXECUTION PATH WITH RACE DETECTION FUNCTION
S230
RACE DETECTION DURING A PROGRAM EXECUTION
S240
FINISH

FIG. 3

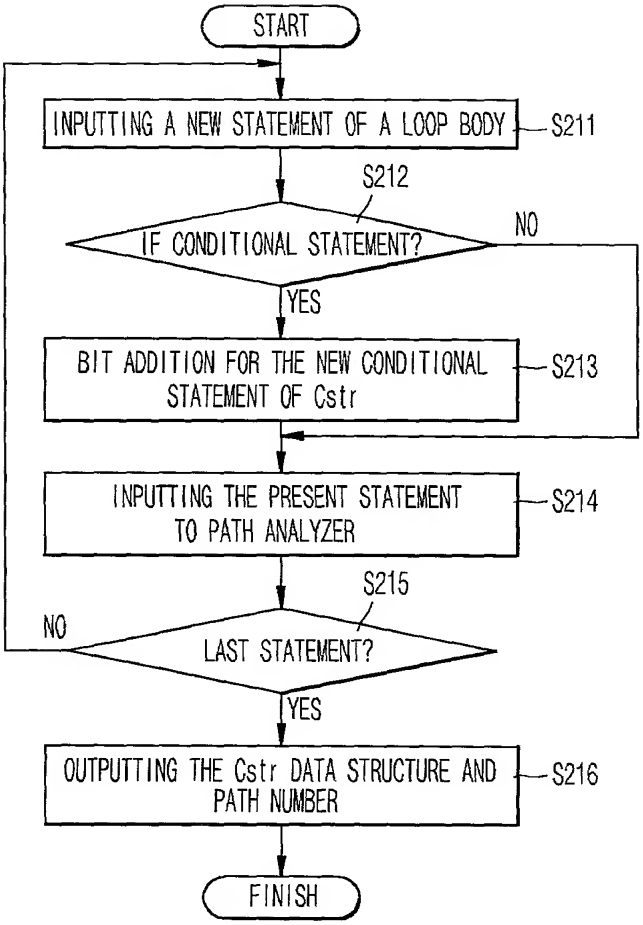

START
INPUTTING A NEW STATEMENT OF A LOOP BODY
S211
S212
IF CONDITIONAL STATEMENT?
NO
YES
BIT ADDITION FOR THE NEW CONDITIONAL STATEMENT OF Cstr
S213
INPUTTING THE PRESENT STATEMENT TO PATH ANALYZER
S214
S215
NO
LAST STATEMENT?
YES
OUTPUTTING THE Cstr DATA STRUCTURE AND PATH NUMBER
S216
FINISH

FIG. 4

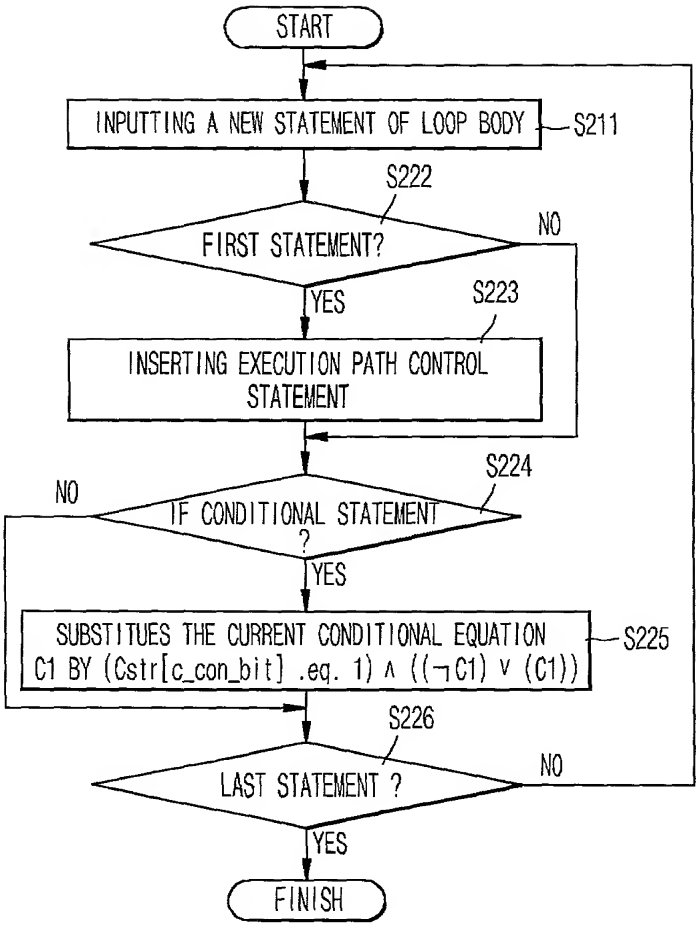
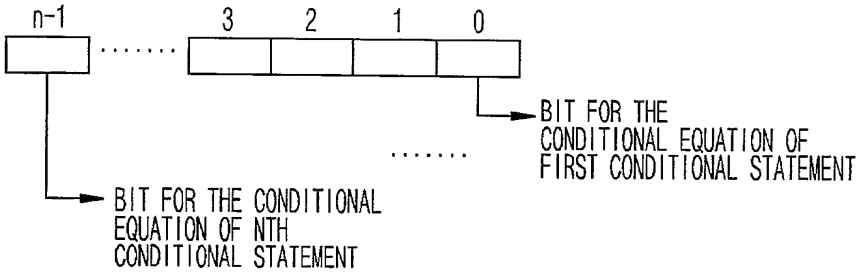

START
INPUTTING A NEW STATEMENT OF LOOP BODY
S211
S222
FIRST STATEMENT?
NO
YES
S223
INSERTING EXECUTION PATH CONTROL STATEMENT
S224
NO
IF CONDITIONAL STATEMENT ?
YES
SUBSTITUES THE CURRENT CONDITIONAL EQUATION C1 BY (Cstr[c_con_bit] .eq. 1) ∧ ((¬C1) ∨ (C1))
S225
S226
LAST STATEMENT ?
NO
YES
FINISH

FIG. 5

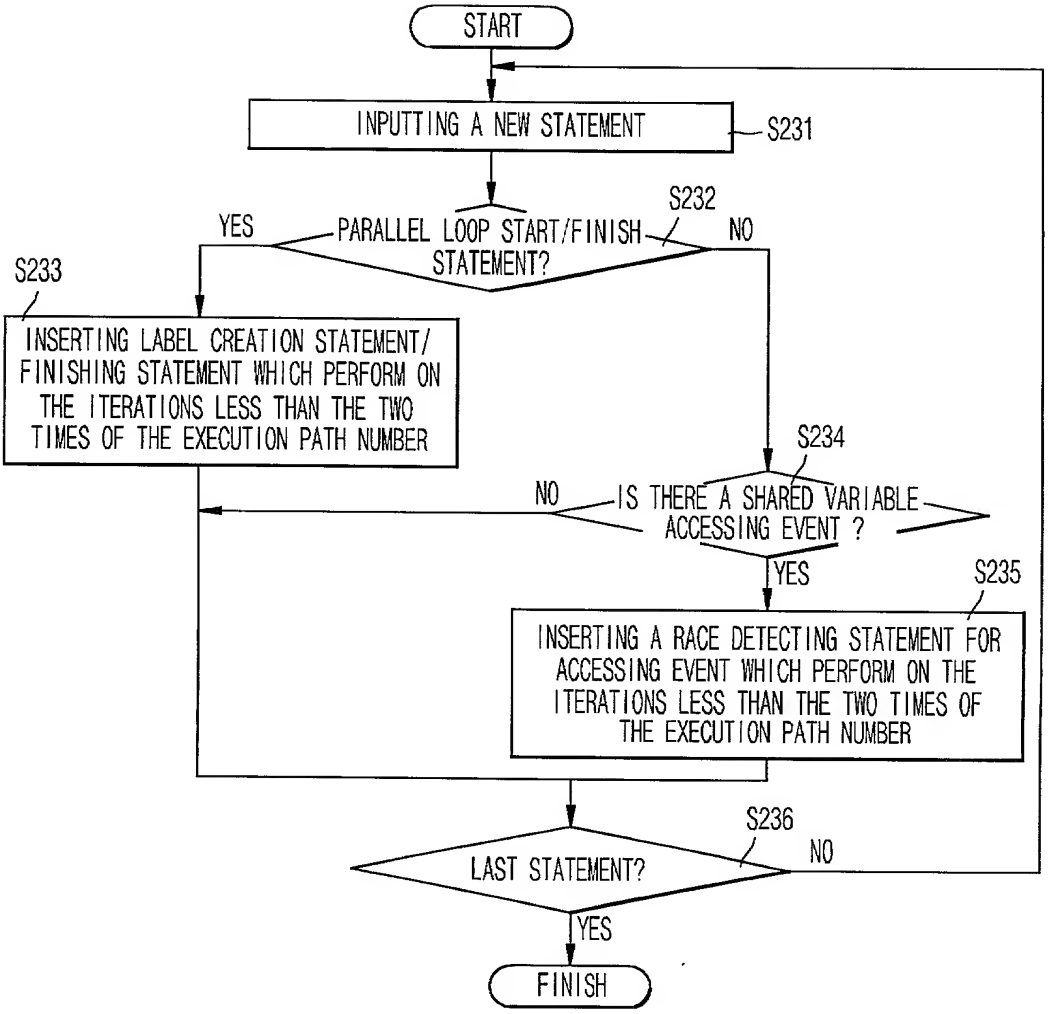
CSTR (CONDITION STATEMENT BRANCH DETERMINANT STRING) CONSTRUCTION OF DATA STRUTURE



THE CONSTRUCTION OF EXECUTION PATH CONTROL STATEMENT

```
P=(I/stride) mod k ;
if (p .eq.0)
    Cstr= (0)2;
else if (p .eq.1)
    Cstr=(1)2;
        :
        :
      else if (p .eq.k-1)
              Cstr=(k-1)2;
endif
,where I is loop control variable
and k is the number of paths in a loop
```

FIG. 6


START
INPUTTING A NEW STATEMENT
S231
S232
PARALLEL LOOP START/FINISH STATEMENT?
YES
NO
S233
INSERTING LABEL CREATION STATEMENT/ FINISHING STATEMENT WHICH PERFORM ON THE ITERATIONS LESS THAN THE TWO TIMES OF THE EXECUTION PATH NUMBER
S234
IS THERE A SHARED VARIABLE ACCESSING EVENT ?
NO
YES
S235
INSERTING A RACE DETECTING STATEMENT FOR ACCESSING EVENT WHICH PERFORM ON THE ITERATIONS LESS THAN THE TWO TIMES OF THE EXECUTION PATH NUMBER
S236
LAST STATEMENT?
NO
YES
FINISH